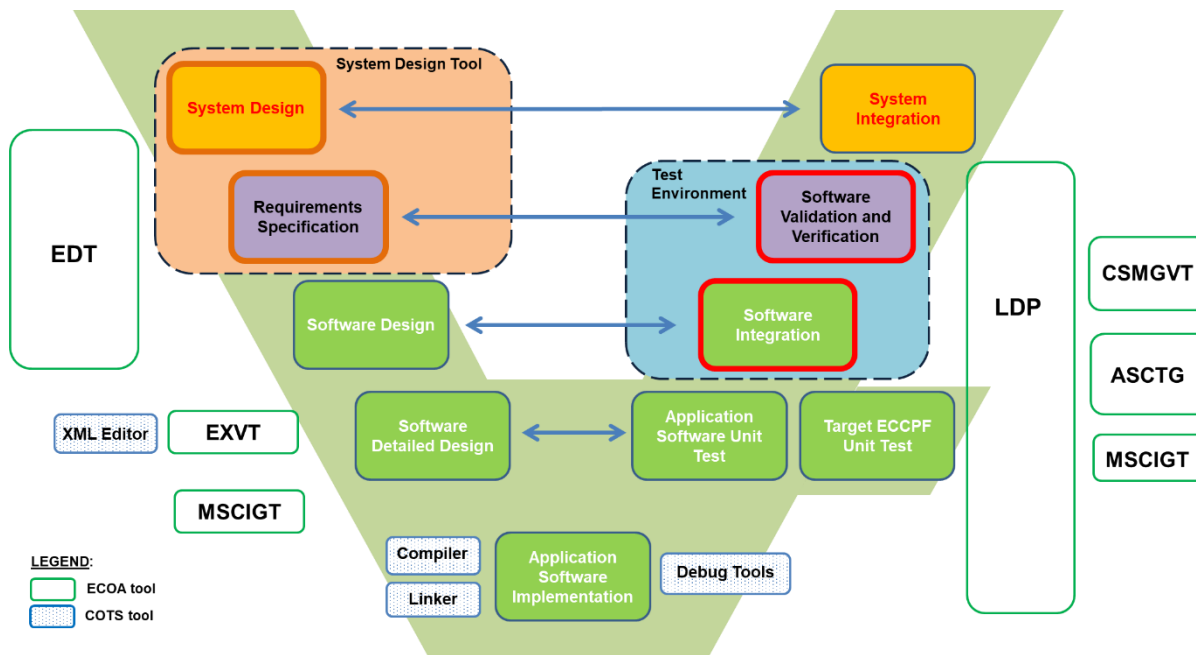


ECOA AS6 TOOLS OVERVIEW

ECO A Tools Overview

Purpose: A set of open source engineering tools has been developed to ease the use of the ECOA Architecture Specification AS6.



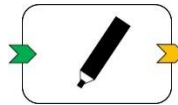
Scope of works

Six main tools:

- **EDT:** ECOA Design Tool
- **MSCIGT:** Module Skeleton and Container Interfaces Generator Tool (including a module-level harness generator)
- **LDP:** Lightweight Development Platform Tool
- **CSMGVT:** Connected System Model Generation and Verification Tool
- **ASCTG:** Application Software Components Test Generator
- **EXVT:** ECOA XML Validation Tool

Presentation of EDT

Alias ECOA EDITOR



Description

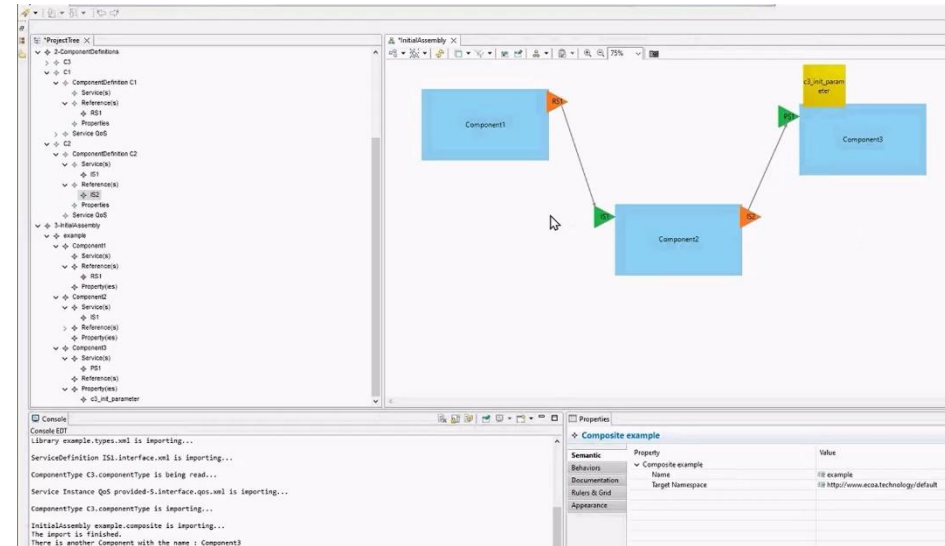
Graphical editor allowing users to create, update or visualize, an ECOA architecture. It offers an AS6 XML import/export function.

Main benefits

- 1.No need to master ECOA meta-model semantic to use the standard.
- 2.Easy appropriation of an ECOA system architecture thanks to user-friendly views.

Who is the tool for ?

Everybody



Environment : Windows

Requires an ECOA validation tool

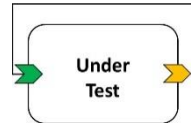
Without such a tool ?

ECOA models have to be manually written with risks of mistakes.

Analysis of XML files required to get architectural views of components assembly or components implementations.

Presentation of ASCTG

Alias ECOA TEST GENERATOR



Description

Generates a new ECOA model describing:

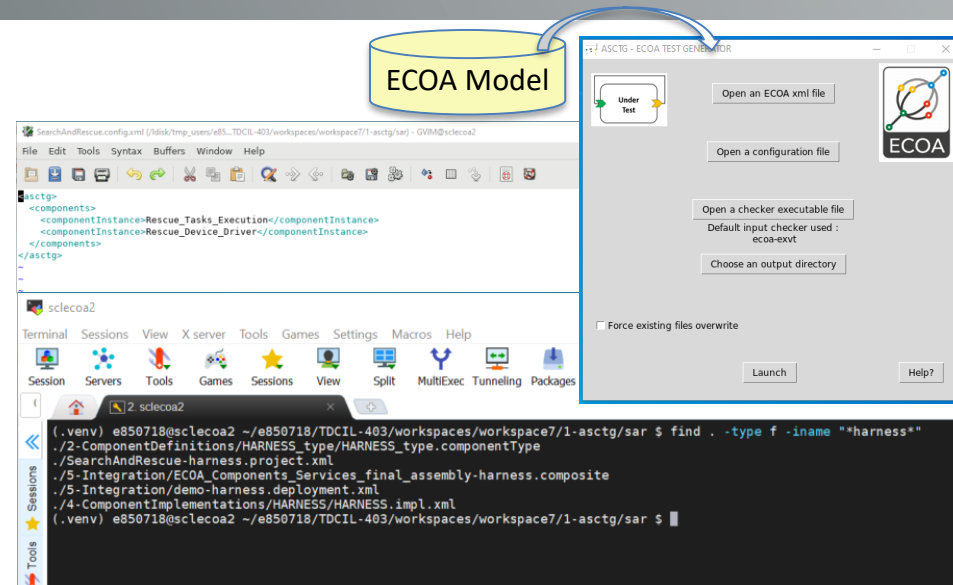
- Components under test + a new “harness” component to stimulate and control them
- New assembly and deployment files for test

Main benefits

Acceleration of components verification process

Who is the tool for ?

Developers and integrators



Environment : Linux

Configuration : ECOA model
+ list of components under test

Requires an ECOA validation tool

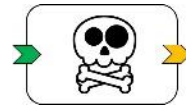
Without such a tool ?

ECOA model to be manually updated to create test artefacts for the chosen subsystem (with risks of mistakes).

Work repeated for each subsystem to be tested.

Presentation of MSCIGT

Alias **ECO**A SKELETON GENERATOR



Description

Generates useful artefacts concerning ECOA modules implementation and test, such as:

- Source code headers and skeletons in accordance with ECOA API,
- Container source code,
- Partial module-level harness source code,
- Makefiles

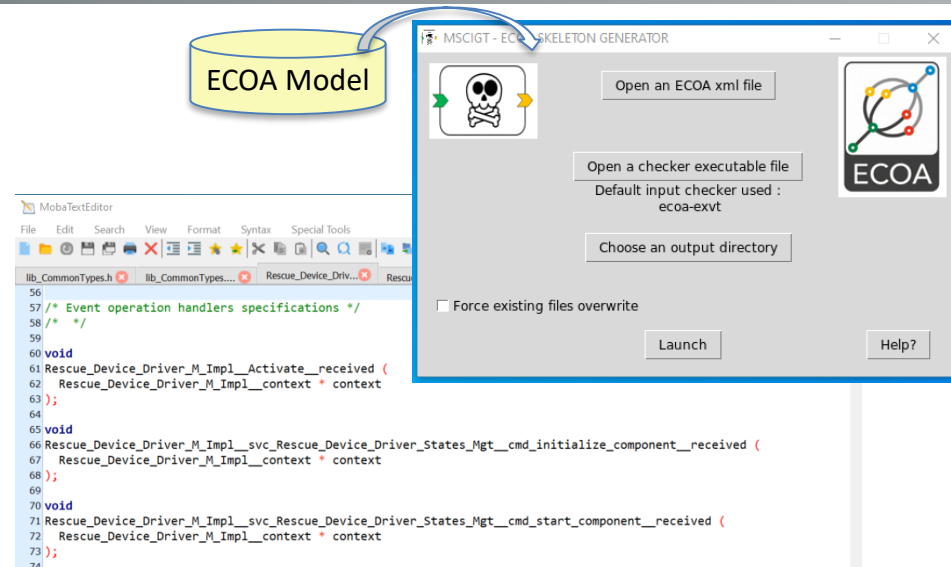
Main benefits

Acceleration of ECOA modules development and test

Who is the tool for ?

Developers

ECO A Model



Environment : Linux

Configuration : ECOA model

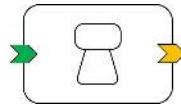
Requires an ECOA validation tool

Without such a tool ?

Corresponding code to be manually written on the basis of ECOA bindings for C and C++ (with risks of mistakes).
In case of evolution of components interfaces, this work is repeated.

Presentation of CSMGVT

Alias ECOA CORK GENERATOR



Description

Allows a non real-time execution of ECOA components, apart from any ECOA middleware, by generating minimal stubs for each API service call. The main purpose of these stubs is to ensure communication between components.

Main benefits

1. Abstracting ECOA middleware to focus components verification on functional behavior,
2. Compliance with office IT environment

Who is the tool for ?

Everybody

ECOA Model

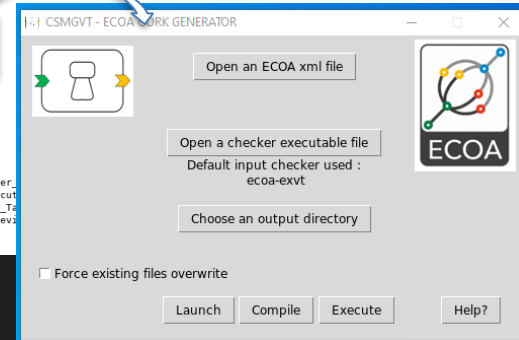
→ Stub code

```
int main(void)
{
    cm_initialize();

    /* Initializing the ECOA modules. */
    Rescue_Device_Driver_M_Impl__INITIALIZE__received(&Rescue_Device_Driver_M_Impl__INITIALIZE__received);
    Rescue_Tasks_Execution_M_Impl__INITIALIZE__received(&Rescue_Tasks_Execution_M_Impl__INITIALIZE__received);
    Rescue_Tasks_Execution_HARNESS_mod_Impl__INITIALIZE__received(&Rescue_Tasks_Execution_HARNESS_mod_Impl__INITIALIZE__received);
    Rescue_Device_Driver_HARNESS_mod_Impl__INITIALIZE__received(&Rescue_Device_Driver_HARNESS_mod_Impl__INITIALIZE__received);

    // Insert arguments logic here.

    [DEBUG] RDD: INITIALIZE
    [DEBUG] RDD: New Component State STOPPED
    [INFO] RTE: RTE (DAV)
    [DEBUG] RTE: INITIALIZE
    [DEBUG] RTE: New Component State STOPPED
    [INFO] RTE_HARNESS: RTE_HARNESS (DAV)
    [DEBUG] RTE_HARNESS: INITIALIZE
    [DEBUG] RTE_HARNESS: New Component State STOPPED
    [INFO] RDD_HARNESS: Rescue Device (DAV)
    [DEBUG] RDD_HARNESS: INITIALIZE
    [DEBUG] RDD_HARNESS: New Component State STOPPED
    [DEBUG] RDD: START
    [DEBUG] RDD: New Component State RUNNING
    [DEBUG] RTE: START
    [DEBUG] RTE: New Component State RUNNING
    [DEBUG] RTE_HARNESS: START
    [DEBUG] RTE_HARNESS: New Component State RUNNING
    [DEBUG] RDD_HARNESS: START
    [DEBUG] RDD_HARNESS: New Component State RUNNING
```



→ Functional execution

Environment : Linux. Compliant with gdb (debug)

Configuration : ECOA model
(deployment is ignored)

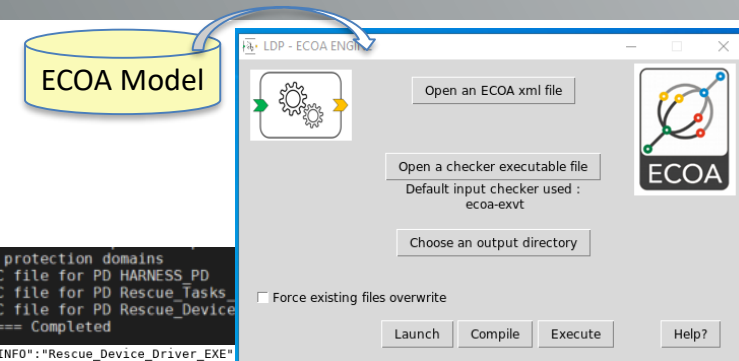
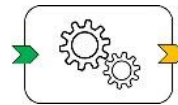
Requires an ECOA validation tool

Without such a tool ?

No possibility to focus on functional test at component level, or manual development of all artefacts (with risks of mistakes).
Work repeated for each subsystem to be tested.

Presentation of LDP

Alias ECOA ENGINE



Description

Generates an ECOA middleware that executes components from an ECOA model, following specified deployment rules. This middleware covers ECOA AS6 core specification, plus ELI, fault handling and graceful shutdown extensions. Only C and C++ bindings are available

Main benefits

1. Ability to execute an ECOA application on wide market means
2. Ability to generate binary files to provide partners with contractual components

Who is the tool for ?

Everybody but mainly developers and integrators

```
INFO == Generate protection domains
INFO # Generate C file for PD HARNESS_PD
INFO # Generate C file for PD Rescue_Tasks
INFO # Generate C file for PD Rescue_Device
INFO ===== Completed
1679565580,035529091":1:"INFO": "Rescue_Device_Driver_EXE"
1679565580,039809480":1:"INFO": "Rescue_Device_Driver_EXE"
1679565580,041791595":1:"INFO": "Rescue_Device_Driver_EXE": "": "thread_name: Rescue_Device_D (1089) - sched_policy:0 - priority:
1679565581,039873213":1:"INFO": "Rescue_Device_Driver_EXE": "": "[Rescue_Device_Driver_EXE] accept new connection on port 20050"
1679565581,040022615":1:"INFO": "Rescue_Device_Driver_EXE": "": "[Rescue_Device_Driver_EXE] accept new connection on port 20054"
1679565581,040254158":1:"INFO": "Rescue_Device_Driver_EXE": "": "[Rescue_Device_Driver_EXE] accept new connection on port 20048"
1679565581,040539347":1:"INFO": "Rescue_Device_Driver_EXE": "": "[Rescue_Device_Driver_EXE]LDP_ID_INIT_MOD received"
1679565581,040615855":1:"INFO": "Rescue_Device_Driver_EXE": "": "[Rescue_Device_Driver_AM] INIT"
1679565581,040644290":1:"INFO": "Rescue_Device_Driver_EXE": "": "[Rescue_Device_Driver_Trigger] INIT trigger"
1679565581,040662257":1:"INFO": "Rescue_Device_Driver_EXE": "": "[Rescue_Device_Driver_Trigger] INIT trigger"
1679565581,049869638":1:"INFO": "Rescue_Device_Driver_EXE": "": "[Rescue_Device_Driver_Trigger] INIT trigger"
1679565581,049887217":1:"INFO": "Rescue_Device_Driver_EXE": "": "[Rescue_Device_Driver_Trigger] INIT trigger"
1679565581,049893175":1:"INFO": "Rescue_Device_Driver_EXE": "": "[Rescue_Device_Driver_Trigger] INIT trigger"
1679565582,050016973":1:"INFO": "Rescue_Device_Driver_EXE": "": "[Rescue_Device_Driver_AM] START"
```

➔ RT execution

Environment : Linux

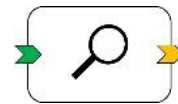
Configuration : ECOA model

Requires an ECOA validation tool

Without such a tool ?
 Another ECOA environment must be found to execute components (maybe in specific expensive benches).
 Manual generation of binary deliveries.

Presentation of EXVT

Alias **ECO CHECKER**



Description

Allows to check both the conformity of a set of XML files with ECOA AS6 specifications, and the consistency of described elements with each other. Can be executed on partial ECOA models.

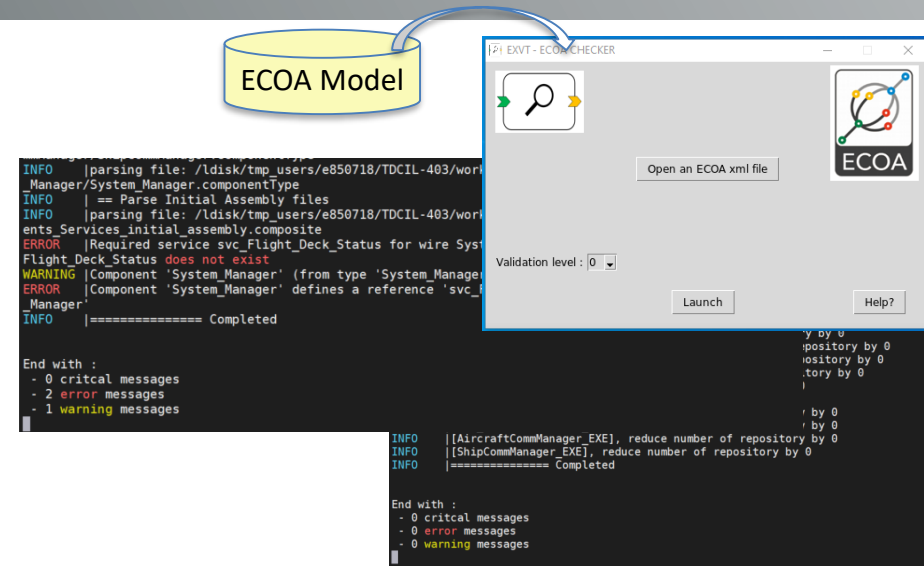
Main benefits

Ensuring the user to own a correct set of ECOA XML files

Who is the tool for ?

Everybody

ECO Model



Environment : Linux

Configuration : ECOA model

Can be used by other tool as the required ECOA validation tool

Without such a tool ?

Risks of mistakes in ECOA models. Components might not be correctly executed in an ECOA environment.

Possible Workflow using Tools

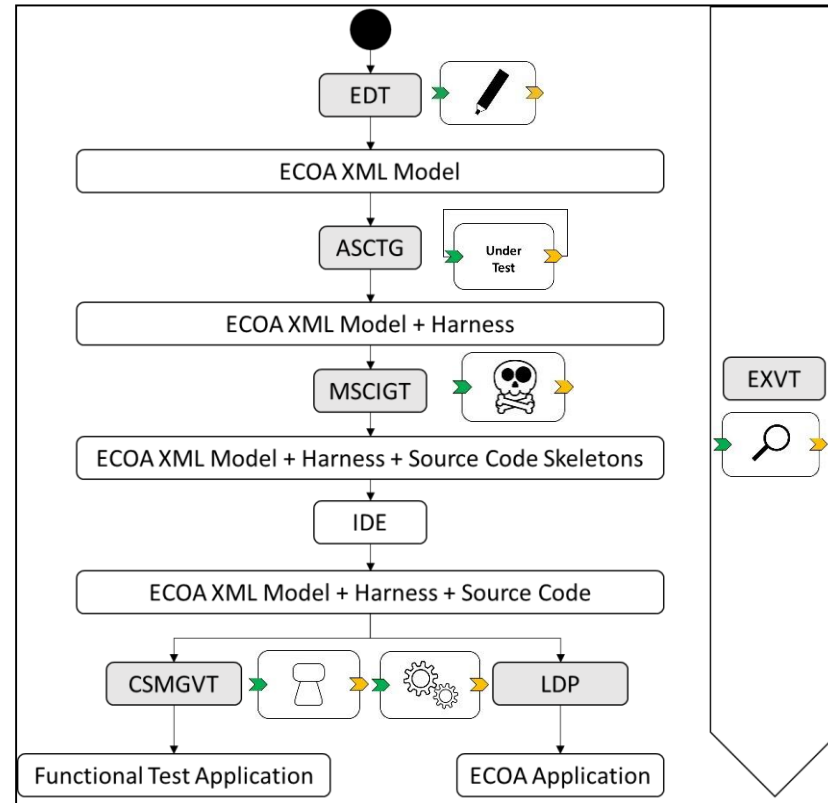
- Begin with EDT to create/import an ECOA model and split the assembly schema in accordance with stakeholders workpackages

For each stakeholder :

- ASCTG to generate component-level test harness
- Component design (breakdown into ECOA modules)
- MSCIGT to generate artefacts for modules implementation and test
 - Development in an IDE (to fill modules and harnesses code skeletons)
 - Modules integration (CSMGVT for functional tests, then LDP for tests in an ECOA environment)

Application-level integration :

- EDT to aggregate partial assemblies including components implementation
- CSMGVT step possible, then LDP for final application test in an ECOA environment



Tool Usage

Tools Launching :

- Direct launching of a tool
 - EDT (*.exe file for windows)
 - Other tools : using a CLI (Command Line Interface) in a Linux terminal

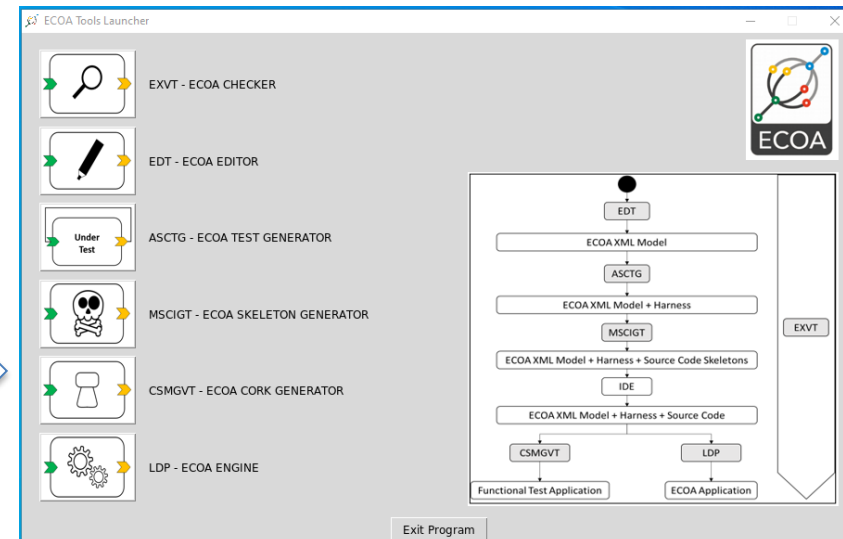
example for asctg :

```
>
> asctg -c ../../example/config.xml -p
../../example.project.xml -k ../exvt
```

- Using ECOA Tools Launcher

Tools outputs :

- All tools aim at generating ECOA artefacts
- Output files are generated in different locations in **ECOA files tree**



Model Data Organisation (according to ECOA AS6)

+ types
source code



+ module
source code



Directory	Sub-directory 1	Sub-directory 2	Sub-directory N	Files	
0-Types	N/A	N/A	N/A	***.types.xml	
1-Services	N/A	N/A	N/A	***.interface.xml	
2-ComponentDefinitions	<name_of_component_definition>	N/A	N/A	<name_of_component>.componentType ***.interface.qos.xml By example: required_<service_name>.interface.qos.xml and provided_<service_name>.interface.qos.xml	
3-InitialAssembly	N/A	N/A	N/A	***.composite	
4-ComponentImplementations	<name_of_implementation>			<name_of_implementation>.impl.xml ***.interface.qos.xml (e.g. new_required_<service_name>.interface.qos.xml) bin-desc.xml Binary files (e.g. *.o or *.dll)	
		0-Dependencies	N/A	Data type, service and component definitions if "0-Types", "1-Services" and "2-ComponentDefinitions" directories are not available.	
		1-Deliverable	N/A	Zipped file of the upper directory	
		Pinfo		Sub-directory for storing Private PINFO	
			<name_of_subdirectory>	Optional Sub-directories for organizing Private PINFO	
5-Integration	N/A			***.impl.composite ***.logical-system.xml ***.deployment.xml sca-contribution.xml ***.cross_platform_view.xml ***.ids.xml	
		0-Dependencies	N/A	N/A	Set of directories containing component implementations if 4-ComponentImplementations is not available
		Pinfo			Sub-directory for storing Public PINFO
			<name_of_subdirectory>	<name_of_subdirectory>	Optional Sub-directories for organizing Public PINFO